

Research Article

Applying LLMs to Active Learning: Toward Cost-Efficient Cross-Task Text Classification Without Manually Labeled Data

Yejian Zhang  and Shingo Takada

Graduate School of Science and Technology, Keio University, Yokohama, Kanagawa, Japan

Correspondence should be addressed to Yejian Zhang; zhangyejian@doi.ics.keio.ac.jp

Received 8 May 2025; Revised 2 June 2025; Accepted 23 June 2025

Academic Editor: Lijie Wen

Copyright © 2025 Yejian Zhang and Shingo Takada. International Journal of Intelligent Systems published by John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

Machine learning-based classifiers have been used for text classification, such as sentiment analysis, news classification, and toxic comment classification. However, supervised machine learning models often require large amounts of labeled data for training, and manual annotation is both labor-intensive and requires domain-specific knowledge, leading to relatively high annotation costs. To address this issue, we propose an approach that integrates large language models (LLMs) into an active learning framework, achieving high cross-task text classification performance without the need for any manually labeled data. Furthermore, compared to directly applying GPT for classification tasks, our approach retains over 93% of its classification performance while requiring only approximately 6% of the computational time and monetary cost, effectively balancing performance and resource efficiency. These findings provide new insights into the efficient utilization of LLMs and active learning algorithms in text classification tasks, paving the way for their broader application.

Keywords: active learning; artificial intelligence; large language models; machine learning; natural language processing; text classification

1. Introduction

Text classification is a fundamental task in natural language processing (NLP), involving the automatic categorization of textual data into predefined categories. It plays a critical role in applications such as sentiment analysis, news classification, and toxic comment detection. These applications span multiple domains and are essential for efficiently processing and analyzing large volumes of text, enabling organizations to extract insights from data. As the volume of textual information grows, the importance of accurate text classification methods increases rapidly, making it a vital component of NLP research and industry applications [1, 2].

Machine learning-based supervised text classifiers have been widely used and have proven effective in performing text classification tasks. However, these models typically require large amounts of labeled data for training, which presents significant challenges. Manual annotation of labeled data is

both labor-intensive and knowledge-intensive, as it requires annotators to have domain-specific expertise to ensure accurate labeling [3]. This process is time-consuming and costly [4, 5], which makes it difficult to scale the development of high-quality training datasets.

The development of large language models (LLMs) has introduced models such as the generative pretrained transformer (GPT), which are capable of performing zero-shot classification tasks through a question-and-answer format. However, these models were not specifically designed for large-scale text classification tasks, which lead to several limitations in such applications [6, 7]. As a large-scale generative model, GPT requires substantial computational resources for each task execution, particularly because classification predictions are made by generating text sequentially through a question-and-answer process [8]. Moreover, inefficiencies in batch processing, delays in model invocation and response times, and the current processing

rate limitations of advanced GPT models further exacerbate the bottlenecks when directly applying GPT to large-scale classification tasks, resulting in higher costs and longer run times [9–11].

To address the aforementioned limitations, we propose a LLM-driven active learning framework. The core idea is to employ a generative LLM (GPT), guided by structured prompts, to annotate instances selected by active learning query strategies, replacing the traditional human oracle. In addition, we evaluate several distinct query strategies and leverage RoBERTa to generate text embeddings that support the query selection process and the training of downstream machine learning classifiers. This design enables our approach to perform cross-task text classification, where “cross-task” refers to handling classification tasks with distinct classification objectives and label spaces (e.g., sentiment analysis, topic categorization, and toxicity detection), without any human-annotated data, while avoiding the disadvantages associated with directly using GPT for large-scale text classification tasks. To validate our approach, we have tested our approach on several datasets including the IMDB dataset [12] for sentiment classification, the AGnews dataset [13] for news categorization, and the Jigsaw Toxic Comment Classification dataset [14] for toxic comment classification.

The main contributions of this work are as follows:

1. We propose a LLM-driven active learning approach for cross-task text classification without relying on human-annotated data.
2. By incorporating GPT into an active learning framework, we mitigate the limitations associated with directly using GPT for large-scale classification tasks, balancing performance and resource efficiency.
3. We demonstrate the effectiveness of our approach through extensive experiments on multiple tasks, including sentiment classification, news classification, and toxic comment classification, achieving high performance across tasks.

2. Related Work

2.1. Importance of Text Classification. Text classification has been widely studied for its critical role in NLP. Sebastiani [15] provided a comprehensive survey of text categorization techniques, highlighting its importance in organizing and managing the growing amount of unstructured text data across various domains. Pang et al. [16] demonstrated the significance of text classification in sentiment analysis, which helps to understand public opinion by automatically determining the sentiment conveyed in reviews or social media posts. Androustopoulos et al. [17] applied text classification to spam detection, showing its effectiveness in filtering unwanted messages. Similarly, Lewis and Gale [18] investigated topic categorization methods for efficient information retrieval, which are critical in enabling better decision-making in fields such as healthcare, finance, and media. These studies demonstrate the role of text classification in automating processes, extracting insights, and managing information efficiently.

2.2. Machine Learning–Based Text Classification. Machine learning methods have been widely applied to text classification. Sebastiani [15] provided an early comprehensive survey on automated text categorization using machine learning, detailing the use of algorithms such as Naive Bayes and support vector machines, and emphasizing the importance of feature extraction techniques such as TF-IDF. Kowsari et al. [19] reviewed traditional machine learning approaches as well as recent deep learning advancements in text classification, highlighting the shift from manual feature engineering to automatic representation learning. Similarly, Zhang and Wallace [20] conducted a survey of text classification techniques, discussing both conventional models and emerging deep learning frameworks, while pointing out the challenges related to labeled data and model scalability. These reviews illustrate the progression of machine learning methods in text classification, while acknowledging the dependency on large labeled datasets.

2.3. LLMs

2.3.1. Bidirectional Transformer–Based Models. Devlin et al. [21] introduced bidirectional encoder representations from transformers (BERT), which employs a bidirectional attention mechanism to capture contextual information from both preceding and following tokens. This bidirectional modeling significantly enhances the quality of learned text embeddings, making them more suitable for a wide range of NLP tasks.

Liu et al. [22] proposed RoBERTa, an optimized variant of BERT, which eliminates the next sentence prediction (NSP) objective and incorporates dynamic masking and larger batch sizes to improve pretraining efficiency. These modifications allow RoBERTa to learn more robust and contextually rich text representations, leading to improved performance in downstream applications. Specifically, RoBERTa has been shown to generate high-quality text embeddings, making it a strong choice for tasks that rely on effective feature extraction rather than generative text modeling.

2.3.2. GPT. Brown et al. [8] introduced GPT-3, a state-of-the-art language model capable of performing various NLP tasks such as text completion, translation, and summarization, with minimal or no task-specific training data. This model exemplifies zero-shot learning capabilities, where extensive pretraining on diverse datasets allows it to generalize across different tasks by leveraging its broad knowledge base rather than explicit supervision.

Building upon GPT-3, OpenAI released GPT-4, a multimodal LLM capable of processing both textual and visual inputs [23]. The model demonstrated strong performance across a broad range of tasks, including complex reasoning, multilingual understanding, and standardized test benchmarks, establishing a new level of general-purpose language intelligence. More recently, GPT-4o was introduced as an optimized version of GPT-4 with enhanced multimodal capabilities and real-time interaction features [24]. It

supports synchronous text, vision, and audio inputs and outputs, enabling fluid and cost-efficient user experiences in interactive applications. These developments illustrate the continuing trend of advancing LLMs toward more general, real-time, and multimodal AI systems.

Alongside these advancements, generative AI technologies have been adopted in a range of practical applications. Notable examples include open-domain question answering [25], news summarization [26], and code generation [27], where generative models support complex reasoning, abstraction, and content creation. These systems are also increasingly integrated into real-world workflows, such as virtual assistants and programming support tools, reflecting the growing presence of generative models in diverse AI-driven applications.

While generative AI models such as GPT have demonstrated impressive performance across multiple domains, their application in structured classification tasks at scale faces certain limitations and challenges. Rae et al. [10] analyzed the computational trade-offs of scaling LLMs, highlighting that while larger models achieve improved performance, they also introduce inefficiencies in inference and deployment. The study shows that GPT-style models, due to their autoregressive nature, require sequential token generation, limiting parallelization and increasing inference latency. Moreover, Raiaa et al. [11] provided a comprehensive analysis of LLMs, highlighting both their advancements and inherent challenges. Their study emphasizes that while models such as GPT demonstrate strong performance across various NLP tasks, their deployment at scale faces computational and efficiency constraints. In addition to discussing the limitations posed by the autoregressive decoding mechanism, they further elaborate on the challenges introduced by high computational cost and energy consumption in real-world applications.

2.4. Active Learning. Active learning has been widely studied as an effective approach to reduce labeling costs by selecting the most informative samples for annotation. Angluin [28] introduced query-based learning in a theoretical framework. Seung et al. [29] proposed query by committee (QBC), where a set of models (a “committee”) is trained on the same labeled data, and the instances that induce the greatest disagreement among them are selected for labeling. Cohn et al. [30] later formalized active learning as a general framework, proposing a query-based selection approach to improve model generalization with fewer labeled examples.

Various query strategies have been introduced to improve active learning effectiveness. Brinker [31] explored the integration of diversity-aware selection into uncertainty-based active learning, ensuring that queried samples are both uncertain and well-distributed in the feature space, thereby reducing redundancy in selected data. Sener and Savarese [32] develop sampling, a geometric approach that formulates active learning as a coverage problem, selecting samples that best represent the entire dataset. Another approach, information density sampling, was introduced by

Settles and Craven [33], where samples are selected based on both uncertainty and density in the feature space, prioritizing representative and high-impact instances.

Active learning has been applied across various domains to enhance model performance while minimizing labeling efforts. Tong and Koller [34] explored the integration of active learning with support vector machines in text classification, demonstrating that selective sampling significantly reduces the number of labeled instances required for accurate classification.

Joshi et al. [35] introduced an active learning approach for multiclass image classification, demonstrating that balancing uncertainty and representativeness improves labeling efficiency in visual recognition tasks. In the realm of structured prediction tasks, Settles and Craven [33] analyzed various active learning strategies for sequence labeling, such as named entity recognition and part-of-speech tagging, highlighting the benefits of selecting informative sequences to improve model accuracy.

3. Proposed Approach

We propose a LLM-driven active learning framework designed to perform cross-task text classification without relying on human-annotated data. The overall workflow of our approach is illustrated in Figure 1. Within the active learning loop, a query strategy selects informative instances from the unlabeled pool \mathcal{U} and sends them to the GPT module for annotation. Guided by structured prompts, the GPT module labels the queried instances Q . The resulting labeled instances $\Delta\mathcal{L}$ are then added to the labeled set \mathcal{L} , which is used to update the machine learning classifier. In the following sections, we detail each component of this loop.

3.1. Query and Update. In the query phase, a query strategy selects a batch of the most informative instances from the current unlabeled pool \mathcal{U} , which are then passed to the oracle for labeling.

In the update phase, the newly labeled instances $\Delta\mathcal{L}$ are added to the labeled set \mathcal{L} and removed from the unlabeled pool \mathcal{U} . The machine learning classifier is retrained on the updated labeled set. Before the first iteration, a seed set is randomly selected from the unlabeled pool \mathcal{U} and labeled by the oracle to initialize the query strategy.

We implemented four widely studied query strategies: uncertainty sampling (QBC) [29], entropy sampling with diversity [3, 18], core-set sampling [32], and information density sampling [33]. Each strategy is evaluated independently in our experiments. We set the logistic regression classifier (solver = lbfgs, multi_class = auto, max_iter = 1000) as the default machine learning classifier, except for QBC, which uses a voting classifier to form the committee. In addition, to support query selection and classifier training, we utilize the RoBERTa-base model with mean pooling to generate fixed-dimensional contextual embeddings. The subsequent subsections provide detailed descriptions of each query strategy.

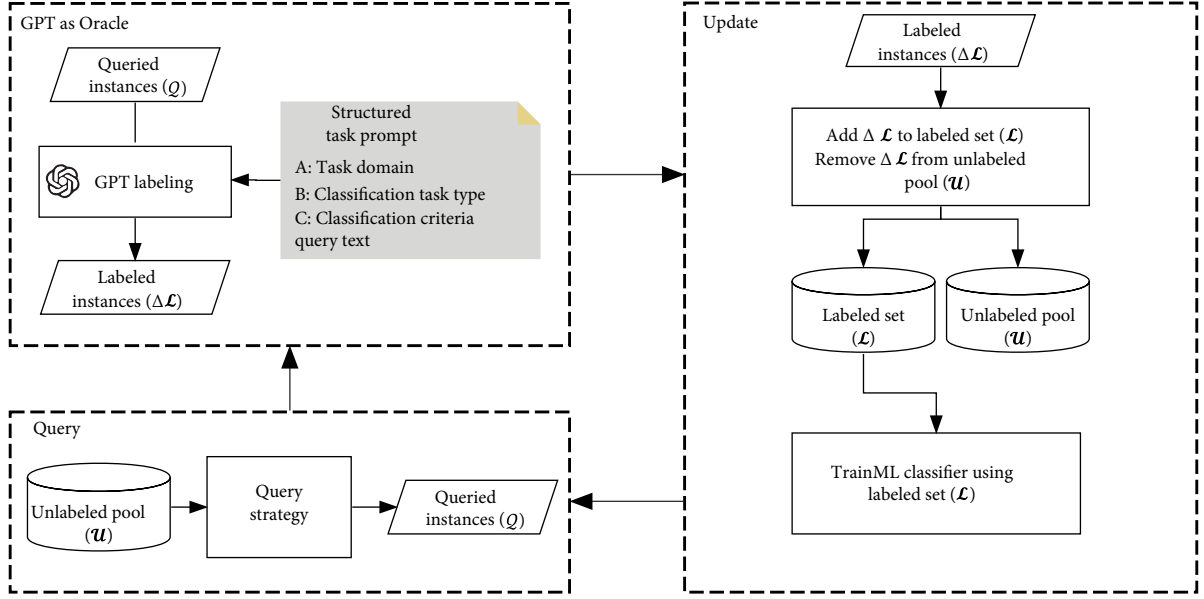


FIGURE 1: Overall workflow: LLM-driven active learning loop.

3.1.1. Uncertainty Sampling (QBC). For uncertainty sampling, we implemented a QBC approach using a voting classifier. The committee consists of four classifiers, including an SVM (Kernel = linear, Probability = True), a decision tree (MaxDepth = None), a random forest (Estimators = 100, MaxFeatures = None), and a logistic regression (solver = lbfgs, multi_class = auto, max_iter = 1000) model. We constructed a soft voting classifier using these four models to better reflect the overall uncertainty, with the calculation of voting entropy given in the following equation:

$$H(x_i) = - \sum_{j=1}^C \left(\frac{1}{M} \sum_{m=1}^M p_{ij}^{(m)} \right) \log \left(\frac{1}{M} \sum_{m=1}^M p_{ij}^{(m)} \right), \quad (1)$$

where $H(x_i)$ is the entropy of sample x_i , C is the total number of classes, M is the number of models in the committee, and $p_{ij}^{(m)}$ is the probability predicted by model m that sample x_i belongs to class j . The samples with the highest voting entropy are selected, representing the instances where the committee models are most uncertain.

3.1.2. Entropy Sampling With Diversity. For entropy sampling with diversity, we use the logistic regression model to predict the probabilities of the unlabeled samples. The entropy for each sample is computed based on the predicted probabilities given in the following equation:

$$H(x_i) = - \sum_{j=1}^C p_{ij} \log p_{ij}, \quad (2)$$

where $H(x_i)$ represents the uncertainty of sample x_i , C is the total number of classes, and p_{ij} is the probability predicted by the logistic regression model that sample x_i belongs to class j . The samples with the highest entropy are initially

selected, representing those with the greatest uncertainty. To ensure diversity among the selected samples, pairwise distances are computed between the high-entropy candidates using RoBERTa embeddings, and diversity scores are assigned based on these distances. The final selection is determined by combining entropy scores and diversity scores for each sample, where the entropy score and diversity score are summed to produce a combined score. The samples with the highest combined scores are selected, resulting in a set that is both uncertain and diverse, thereby improving the model's generalizability by covering a broader region of the input space.

3.1.3. Core-Set Sampling. For core-set sampling, we use the precomputed embeddings from the RoBERTa embedder to map samples into a high-dimensional feature space. Pairwise distances are then computed between the unlabeled samples and the already labeled samples to determine the minimum distance for each unlabeled sample. The goal is to select samples that are farthest from the labeled set, thereby ensuring coverage of different regions of the feature space. Specifically, the samples with the largest minimum distance are chosen, allowing the model to expand its understanding by exploring less-represented areas of the data distribution.

3.1.4. Information Density Sampling. For information density sampling, we aim to select samples that are both uncertain and located in densely populated regions of the feature space. We use the same logistic regression model as in entropy sampling with diversity to predict the probabilities of all the unlabeled samples and compute the entropy of each sample to quantify its uncertainty. To incorporate information density, we also compute the pairwise distances between each unlabeled sample and all other unlabeled samples, assigning a density score based on how densely

a sample is surrounded by other samples. The final score for each sample is determined by multiplying the entropy score and the density score, ensuring that the selected samples are informative and representative of dense regions in the feature space.

3.2. GPT as Oracle. GPT is used as an oracle in the active learning loop to provide labels for the queried instances. By leveraging GPT’s extensive pretrained knowledge, the need for human annotators is eliminated. To achieve this, prompt engineering is utilized to harness the question–answering capabilities of the GPT-4o model to issue labels. To ensure deterministic behavior and concise outputs suitable for labeling, we configured the GPT-4o model with temperature=0 and max_tokens=10 in all API calls. To ensure applicability across different tasks, we designed a structured prompt, as shown in Table 1. The prompt consists of several components: “A” specifies the classification task, “B” indicates whether it is a binary or multiclass classification, and “C” defines the classification standard. Finally, “query” contains the natural language text of instances selected by the active learning strategy. In conclusion, this design not only reduces reliance on human experts but also mitigates the limitations of using GPT directly for large-scale classification, such as high computation costs and long processing times.

4. Evaluation

4.1. Research Questions. To evaluate our approach, we focus on the following research questions:

- RQ1: How does our approach perform across different tasks without using any human-labeled samples?
- RQ2: What is the impact of using GPT as an oracle in active learning, compared to traditional human expert labeling?
- RQ3: What are the advantages of our approach compared to directly using GPT for classification tasks?
- RQ4: What is the contribution of the active learning component in our approach?

4.2. Data Preparation. To evaluate our approach, we prepared three widely used datasets for three different text classification tasks: the IMDB movie reviews dataset for sentiment classification, the AGnews dataset for news categorization, and the Jigsaw Toxic Comment Classification dataset for the toxic comment classification task.

First, we preprocessed all three datasets to standardize the labels. In the IMDB dataset, “1” was used to represent positive sentiment, and “0” for negative sentiment. In the AGnews dataset, labels were assigned as follows: “0” for world, “1” for sports, “2” for business, and “3” for sci/tech. For the Jigsaw Toxic Comment Classification dataset, “0” was used for nontoxic comments and “1” for toxic comments.

TABLE 1: GPT prompt structure.

Structured prompt
{“role”: “system”, “content”: f“You are an expert in {A}.”},
{“role”: “user”, “content”: f“Now you have a {B}.
Please {C}:{query}”. Please only return the label.”}

After label standardization, we randomly selected 10,000 instances from both the IMDB and AGnews datasets for our experiments. As for the Jigsaw Toxic Comment Classification dataset, since it had issues such as data with very few words and significant class imbalance, we further processed the dataset by removing extremely short instances, specifically, those with five words or fewer. After this filtering, we selected 5000 instances each from toxic and nontoxic comments to create the final experimental dataset. The text length distributions of the three datasets are shown in Figures 2, 3, and 4. The label distributions are summarized in Figure 5.

4.3. Experimental Setup. The implementation details of our approach have been discussed in Section 3. This section focuses on the evaluation metrics, experimental environment, and structured prompt settings for the experiments based on the research questions.

4.3.1. Evaluation Metrics. We used accuracy as the primary evaluation metric, along with F1 score and recall to assess the performance of our experiments. The calculation for accuracy is given in equation (3). The F1 score for binary classification is calculated as shown in equation (4), and for multiclass classification, macro F1 is given by equation (5). Similarly, recall for binary classification is computed as in equation (6), and macro recall for multiclass classification is calculated as in equation (7).

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}, \quad (3)$$

$$\text{F1 (binary)} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}, \quad (4)$$

$$\text{Macro F1} = \frac{1}{C} \sum_{i=1}^C \frac{2 \times \text{TP}_i}{2 \times \text{TP}_i + \text{FP}_i + \text{FN}_i}, \quad (5)$$

$$\text{Recall (binary)} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (6)$$

$$\text{Macro recall} = \frac{1}{C} \sum_{i=1}^C \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}, \quad (7)$$

where TP (true positive) is the number of correctly predicted positive instances, FP (false positive) is the number of incorrectly predicted positive instances, and FN (false negative) is the number of missed positive instances. In the case of multiclass classification, C represents the total number of

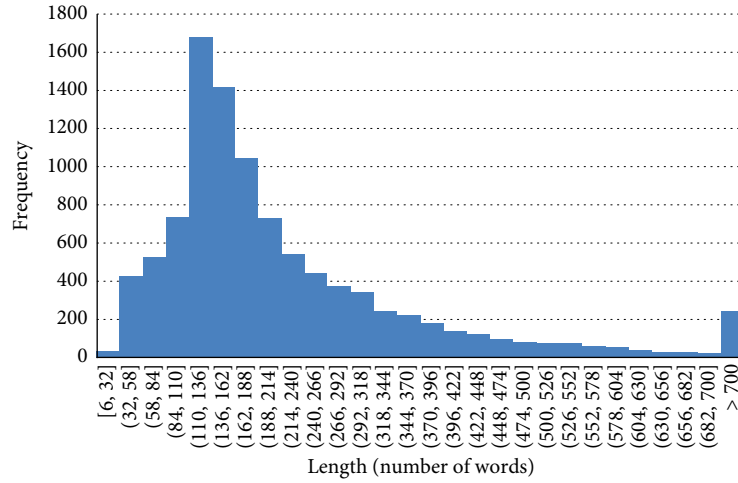


FIGURE 2: Length distribution of the IMDB dataset.

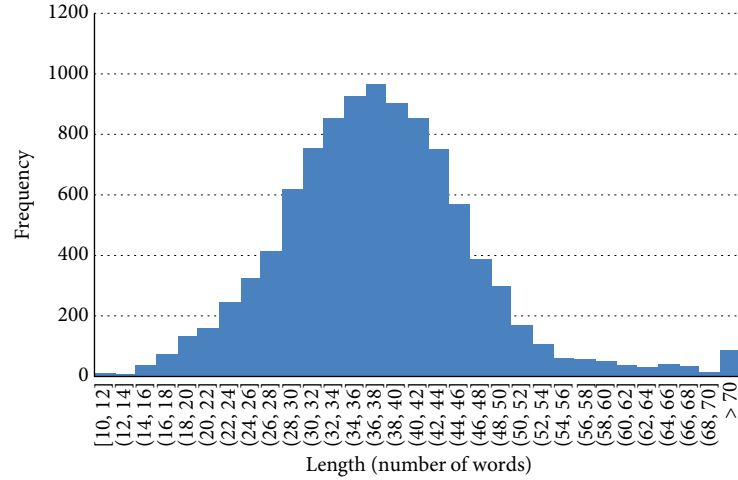


FIGURE 3: Length distribution of the AGnews dataset.

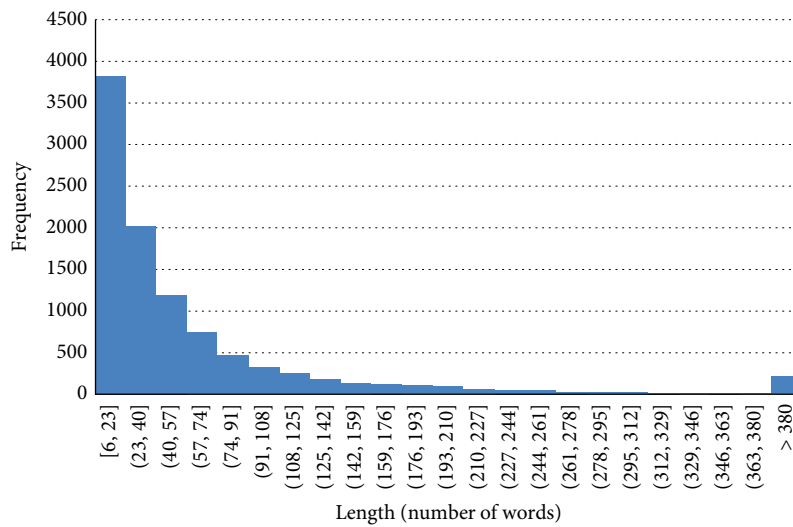


FIGURE 4: Length distribution of the Jigsaw Toxic Comment Classification dataset.

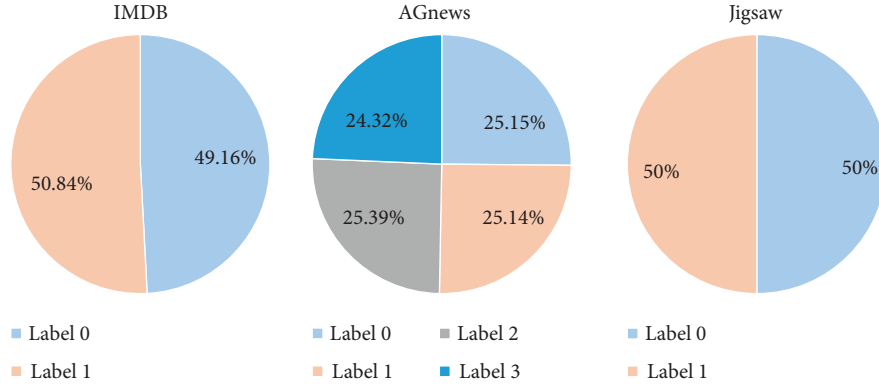


FIGURE 5: Label distributions.

classes, and TP_i , FP_i , and FN_i correspond to the counts for class i .

4.3.2. Experimental Environment. All experiments were conducted in the Google Colab environment, utilizing an NVIDIA A100-SXM4-40GB and an Intel(R) Xeon(R) CPU @ 2.20 GHz.

4.3.3. Structured Prompt Settings. Based on the three datasets prepared for the classification tasks described in Section 4.2, we populated the structured prompt Components A, B, and C, as shown in Table 1 of Section 3.2, for each task as follows:

- IMDB sentiment classification task:
 - A = “user reviews sentiment classification”
 - B = “binary sentiment classification task”
 - C = “classify the following user review into positive sentiment or negative sentiment, use 1 for positive and 0 for negative”
- AGnews news classification task:
 - A = “news article classification”
 - B = “four-class news topic classification task”
 - C = “classify the following news article into one of the following categories: 0 for World, 1 for Sports, 2 for Business, or 3 for Sci/Tech”
- Jigsaw Toxic Comment Classification task:
 - A = “toxic comment classification”
 - B = “binary classification task”
 - C = “classify the following comment into toxic or nontoxic, use 1 for toxic and 0 for non-toxic”

A complete example of a structured prompt along with the corresponding GPT output is provided in Appendix A to illustrate the actual query-response interaction.

4.4. Evaluation for RQ1. To address RQ1, we used all three datasets and evaluated our approach primarily using accuracy, with F1 score and recall as supplementary metrics. To ensure the robustness and reliability of the experimental

results, we performed 5-fold cross-validation, with each fold using 80% of the dataset for training and 20% for testing.

In each fold of the cross-validation, we initially randomly selected 50 instances (representing 0.625% of the training set) from the training set as the seed set for starting active learning and used structured prompts to request labels from GPT. We then began the active learning loop, setting the batch size to 5 and the number of iterations to 100. This means that each active learning loop consisted of 100 iterations, with each of the four query strategies independently selecting the top 5 most uncertain instances in each iteration and requesting labels from GPT using our structured prompts. The obtained labels were then used to update each independent model.

Figures 6, 7, and 8, respectively, show the learning curve of our approach for the IMDB sentiment, AGnews, and Jigsaw Toxic Comment Classification tasks. The x -axis represents the number of iterations, and the y -axis represents the primary evaluation metric, accuracy, which is the average accuracy obtained from 5-fold cross-validation. The four lines represent the four query strategies combined with GPT.

From Figure 6, we observed that for the IMDB sentiment classification task, the combination of GPT with entropy sampling with diversity performed the best. Since active learning involves a trade-off between performance and labeling cost, we chose the 25th iteration, when the performance improvement began to plateau, as the final result for the sentiment classification task on the IMDB dataset, as recorded in Table 2.

From Figure 7, we observed that for the AGnews news classification task, both information density and entropy sampling with diversity demonstrated similar strong performance. We chose the data from the 30th iteration, where the performance growth of both methods began to plateau, and selected the higher performance (with information density showing a slight advantage at that point) as the final performance of our approach for the news classification task. The detailed results are recorded in Table 3.

From Figure 8, we observed that for the Jigsaw Toxic Comment Classification task, entropy sampling with diversity demonstrated an advantage. Similarly, since active learning involves a trade-off between labeling cost and

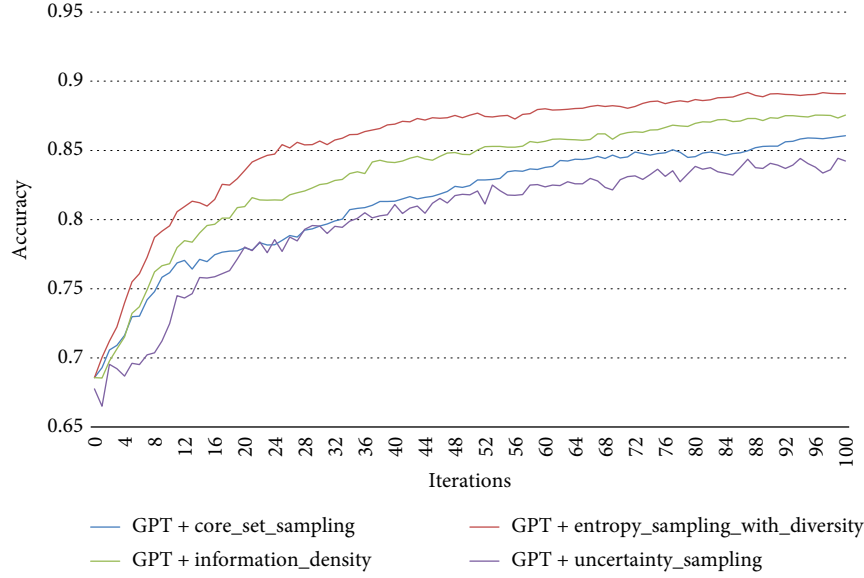


FIGURE 6: RQ1 learning curve: IMDB sentiment classification.

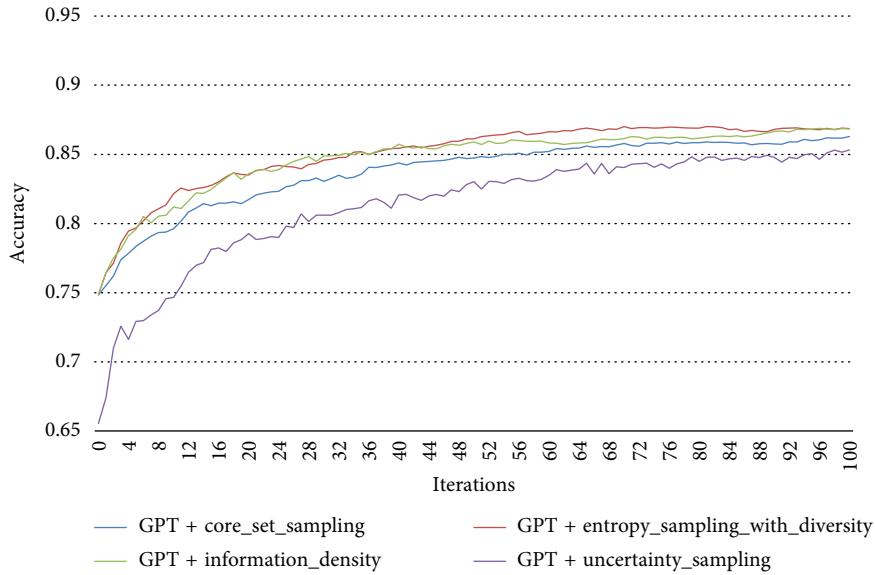


FIGURE 7: RQ1 learning curve: AGnews news classification.

performance, we selected the result from the 19th iteration, where the performance growth began to plateau, as the final toxic comment classification performance of our approach. The detailed results are presented in Table 4.

In summary, our approach achieved notable classification performance without any human-labeled data. Specifically, on the IMDB sentiment classification task, our method obtained an accuracy of 85.42% by using GPT for labels on 175 instances. For the AGnews news classification task, an accuracy of 84.88% was achieved by using 200 labeled instances from GPT. Lastly, for the Jigsaw Toxic Comment Classification task, an accuracy of 86.44% was obtained by using 145 labeled instances from GPT. These results demonstrate that our approach effectively achieves

high classification performance across different tasks without relying on human annotations, proving both its effectiveness and its cross-task generalization capability.

4.5. Evaluation for RQ2

4.5.1. RQ2 Setup. To evaluate the impact of GPT as an oracle in active learning, we compared the following three scenarios:

1. Full GPT labeling: GPT is used for labeling both the seed set and all queries.
2. Hybrid labeling: Human annotators provide labels for the seed set, while GPT labels all queries.

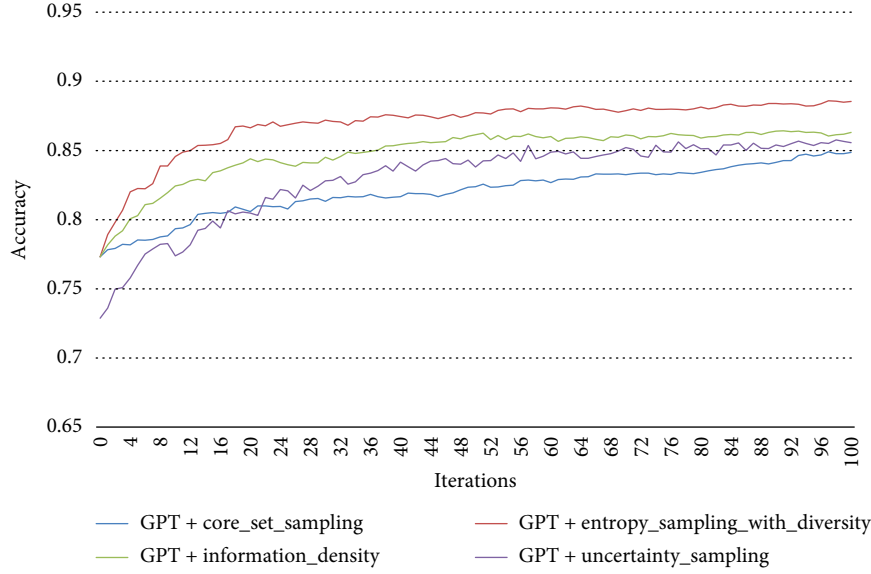


FIGURE 8: RQ1 learning curve: Jigsaw Toxic Comment Classification.

TABLE 2: RQ1 result: IMDB sentiment classification.

Fold	Accuracy	F1	Recall
1	0.8555	0.8555	0.8557
2	0.8685	0.8680	0.8683
3	0.8335	0.8335	0.8345
4	0.8635	0.8635	0.8638
5	0.8500	0.8497	0.8509
Average	0.8542	0.8540	0.8546

Note: The bold context in the last row is intended to help readers quickly identify the overall average metrics.

TABLE 3: RQ1 result: AGnews news classification.

Fold	Accuracy	F1	Recall
1	0.8345	0.8347	0.8351
2	0.8285	0.8252	0.8258
3	0.8500	0.8486	0.8495
4	0.8695	0.8680	0.8684
5	0.8615	0.8610	0.8604
Average	0.8488	0.8475	0.8478

Note: The bold context in the last row is intended to help readers quickly identify the overall average metrics.

TABLE 4: RQ1 result: Jigsaw Toxic Comment Classification.

Fold	Accuracy	F1	Recall
1	0.8450	0.8442	0.8449
2	0.8665	0.8665	0.8666
3	0.8705	0.8704	0.8704
4	0.8725	0.8721	0.8721
5	0.8675	0.8674	0.8679
Average	0.8644	0.8641	0.8644

Note: The bold context in the last row is intended to help readers quickly identify the overall average metrics.

3. Full human labeling: Human annotators label both the seed set and all queries.

For scenarios requiring human-labeled samples, we directly used the labels from the original dataset. To ensure

fairness in the comparative experiments, we used the same 5-fold splits and conducted experiments on the three classification tasks using the same experimental setup as in RQ1.

4.5.2. RQ2 Results. Figures 9, 10, and 11, respectively, show the comparative learning curves for the four query strategies on the IMDB sentiment, AGnews, and Jigsaw Toxic Comment Classification task. Similar to RQ1, the y-axis represents the average accuracy from the 5-fold cross-validation, while the x-axis represents the number of iterations. Each figure contains four subplots, each representing a corresponding query strategy. In each subplot, three lines are shown, representing the cases of full GPT labeling, hybrid labeling, and full human labeling.

To provide a more detailed understanding of the comparative learning curves, we first calculated the standard deviation of the accuracy obtained from full GPT labeling, full human labeling, and hybrid labeling at each iteration. Then, we averaged these standard deviations across all iterations within each query strategy to obtain the average standard deviation for each query strategy in each of the three classification tasks. Detailed standard deviation results are listed in Table 5. For each query strategy on each task, the maximum cross-iteration average standard deviation observed was 0.007961, and the overall average standard deviation across all tasks and query strategies was 0.004056. The relatively small average standard deviation across all query strategies and tasks indicates that the accuracy differences among full human labeling, hybrid labeling, and full GPT labeling remain stable across iterations.

Furthermore, we calculated the accuracy differences between GPT labeling and full human labeling, as well as between GPT labeling and partial human labeling, at each iteration. We then averaged these accuracy differences across all iterations within each query strategy to obtain the average accuracy difference for each query strategy in each of the three classification tasks. Detailed accuracy difference results

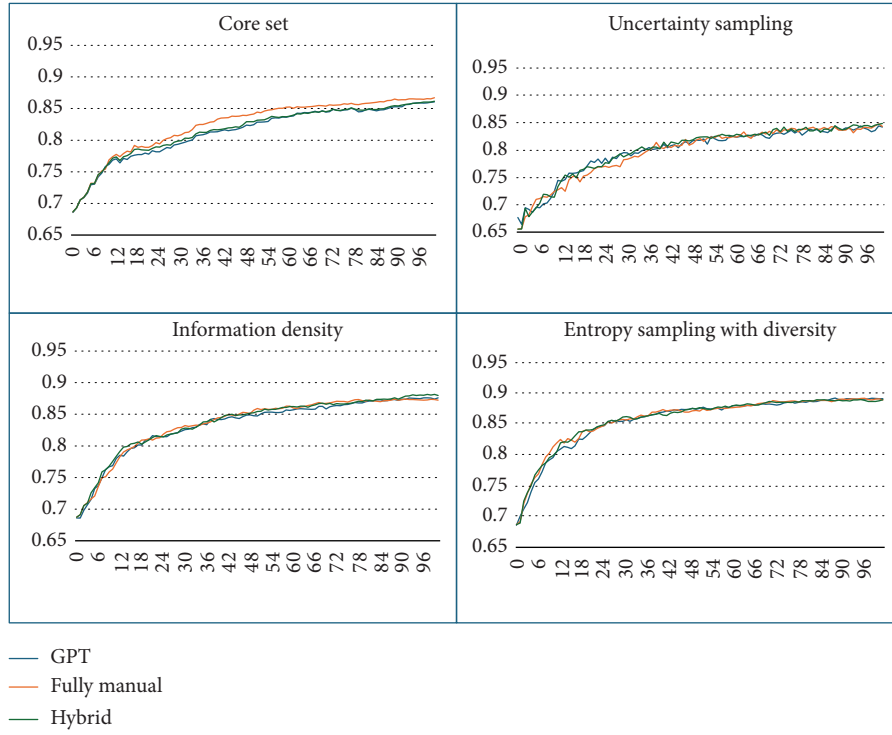


FIGURE 9: RQ2 learning curves: IMDB sentiment classification.

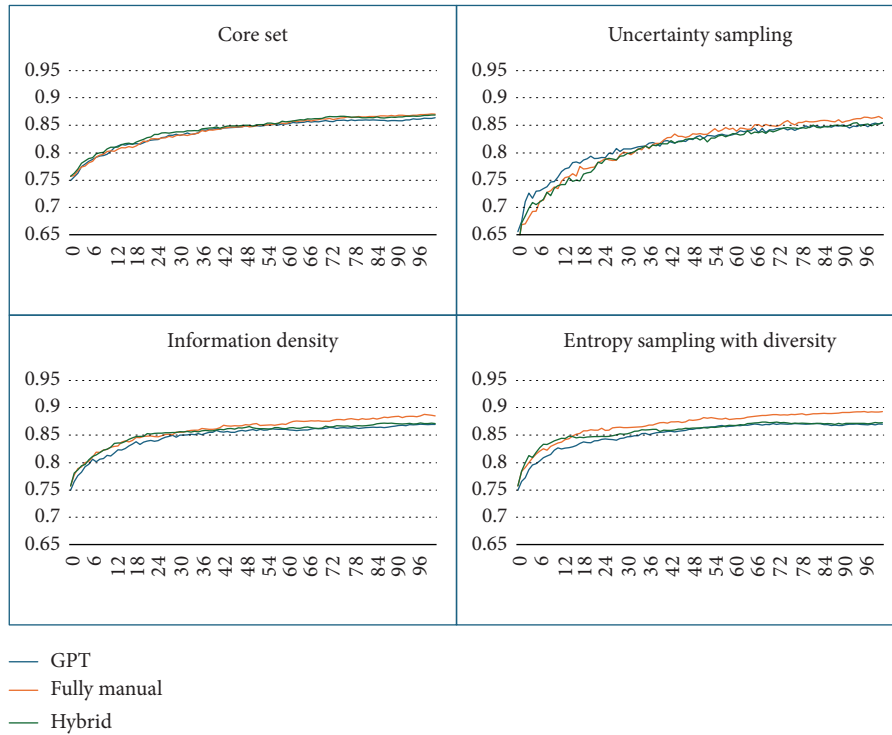


FIGURE 10: RQ2 learning curves: AGnews news classification.

are shown in Table 6, where negative values indicate that the accuracy of the active learning loop using full GPT labeling was lower, whereas positive values indicate higher accuracy. The worst case for full GPT labeling against full human

labeling was -1.7321% , while the average across all tasks and strategies was -0.5168% . Similarly, the worst case of using full GPT labeling against hybrid labeling was -0.5985% , and the overall average was -0.1264% . These results demonstrate

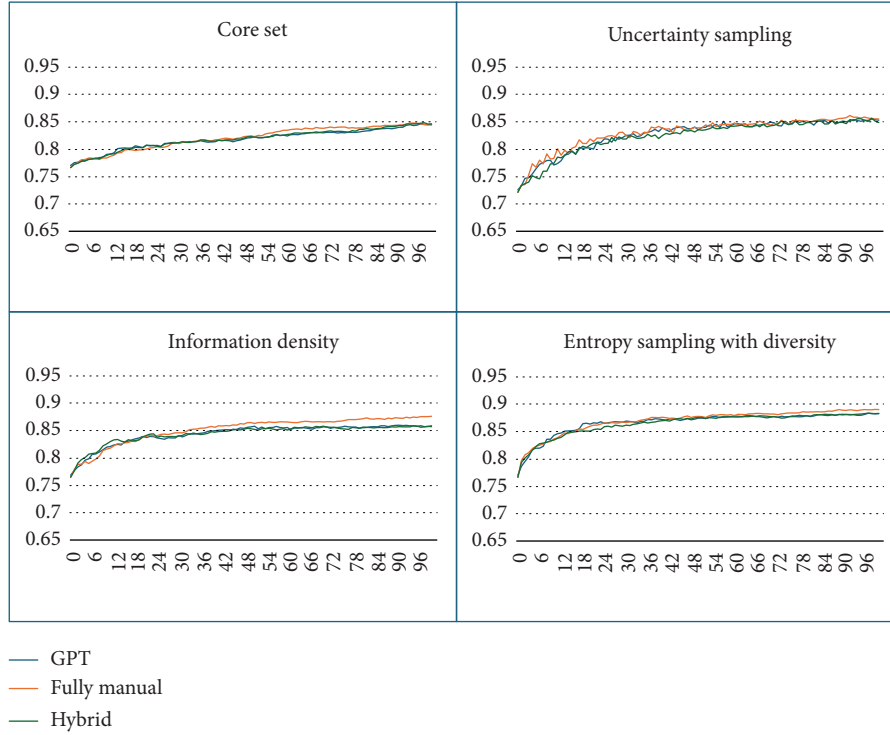


FIGURE 11: RQ2 learning curves: Jigsaw Toxic Comment Classification.

TABLE 5: RQ2 result: Average standard deviation.

	Core set	Entropy & diversity	Information	Uncertainty
IMDB	0.004939	0.002142	0.002463	0.003870
AGnews	0.002546	0.007961	0.005305	0.005883
Toxic	0.002156	0.002818	0.004879	0.003715

TABLE 6: RQ2 result: Comparison of average accuracy differences.

		Core set	Entropy and diversity	Information	Uncertainty
GPT versus fully manual	IMDB	-0.011189	-0.002010	-0.002209	0.001413
	AGnews	-0.002059	-0.017321	-0.011646	0.000159
	Toxic	-0.002556	-0.003018	-0.007780	-0.003796
GPT versus semimanual	IMDB	-0.002429	-0.001682	-0.003287	-0.001831
	AGnews	-0.004733	-0.005985	-0.005905	0.006666
	Toxic	-0.000639	0.001758	-0.000205	0.003101

that, despite minor fluctuations, the overall accuracy disadvantage of full GPT labeling is also small, reinforcing the observation that GPT labeling performs comparably to both human labeling and hybrid labeling across the evaluated tasks.

4.5.3. RQ2 Conclusion. Based on these results, we conclude that there is no substantial difference in the overall performance of active learning when using full human labeling, hybrid labeling, or fully GPT-labeled data.

It is also important to note that in simulating human labeling, we used the ground truth labels from the original datasets. Since these datasets are well-known and have been

widely used in academic research, we assume that the labeling quality is high. In other labeling scenarios, the quality of human labeling might not reach the same level of rigor.

In conclusion, given the comparable performance between using GPT labeling and human labeling for active learning, GPT labeling offers a viable and cost-effective alternative for large-scale annotation tasks, particularly when human resources are limited.

4.6. Evaluation for RQ3. To address RQ3, we evaluated the efficiency and cost-effectiveness of our proposed approach by comparing it with a baseline where GPT is directly used as a classifier without any active learning or model training.

TABLE 7: RQ3 result: Comparison between our approach and direct GPT usage.

	Direct GPT usage			Our approach		
	Accuracy	Time (min)	Cost (USD)	Accuracy	Time (min)	Cost (USD)
IMDB	0.9463	17.81	3.23	0.8542	1.29	0.20
AGnews	0.8866	15.14	1.17	0.8488	0.89	0.09
Jigsaw	0.9178	16.20	1.58	0.8644	0.80	0.06

TABLE 8: RQ4 result: Performance without active learning.

	Fold1	Fold2	Fold3	Fold4	Fold5	Average
IMDB	0.7515	0.7495	0.8010	0.7880	0.7880	0.7756
AGnews	0.7840	0.7945	0.7995	0.8140	0.8225	0.8029
Toxic	0.8000	0.8045	0.7880	0.8010	0.8075	0.8002

Specifically, we employed the structured prompt described in Section 4.3.3 to obtain predictions for the test set.

To ensure fairness in the comparative experiments, we used the same 5-fold cross-validation split as in RQ1. For each dataset, we recorded the following metrics:

- Classification accuracy: the average accuracy across five folds.
- Time expenditure: The average classification time across five folds.
- Monetary cost: the average cost of GPT API usage across five folds, calculated based on the number of tokens processed.

Table 7 summarizes the 5-fold average classification accuracy, time expenditure, and cost for both direct GPT usage and our proposed approach across the three datasets.

Based on the experimental results, our approach required only 6% of the time and monetary cost compared to directly using GPT for classification, while still achieving over 93% of its classification performance. These findings demonstrate that our approach is not only significantly more cost-effective but also highly efficient, making it particularly advantageous for cost-sensitive applications.

4.7. Evaluation for RQ4. To answer RQ4, we disabled the query strategy components and instead randomly selected the same number of instances from the training set as the combined total of the seed and query instances used in each of the final performance settings of RQ1. We used the sampled instances to train the default logistic regression classifier from the active learning module. To ensure the rigor of the comparative experiment, we conducted the same 5-fold cross-validation, and for each fold, we repeated the random sampling five times, taking the average as the result for that fold. The detailed accuracy results are shown in Table 8.

As observed, incorporating the active learning component led to an average accuracy improvement of 7.86% in the IMDB sentiment classification task, 4.59% in the AGnews news classification task, and 6.42% in the Jigsaw Toxic

Comment Classification task. This demonstrates that active learning contributes to model performance by guiding the selection of informative instances. Therefore, we conclude that integrating active learning into our approach is beneficial, as it enhances classification accuracy with a limited number of labeled instances.

5. Conclusion

This study proposes a cost-efficient text classification approach that integrates LLMs with active learning, enabling effective cross-task text classification without requiring manually labeled data. Experimental results confirm that our approach successfully eliminates the need for human annotations while ensuring high classification performance across diverse tasks.

Furthermore, we systematically evaluated the role of GPT as an oracle in active learning and quantified the contribution of different query strategies within the overall framework. Moreover, our experiments demonstrate that compared to directly using generative AI models such as GPT for text classification, our approach achieves comparable performance while significantly reducing the cost.

In terms of practical implications, our approach can be particularly valuable in resource-constrained scenarios where labeled data are scarce or manual annotation is infeasible. To evaluate its generalizability, we conducted experiments on three distinct classification tasks: sentiment analysis, topic categorization, and toxicity detection. These tasks differ in their semantic focus and label spaces, providing initial evidence of the approach’s robustness across task types. This demonstrated cross-task generalization capability broadens the potential applicability of our approach to a wide range of text classification scenarios. However, the current evaluation has certain limitations, as it is restricted to English-language data and uses only one dataset per task.

Future work could extend the approach to more diverse domains and multilingual data. Such investigations would help more comprehensively demonstrate the approach’s adaptability and deployment potential across broader application scenarios.

Appendix A: Example of Structured Prompt and Output

Input Prompt:

```
{“role”: “system”, “content”: “You are an expert in user reviews sentiment classification.”}
{“role”: “user”, “content”: “Now you have a binary sentiment classification task. Please classify the following user review into positive sentiment or negative sentiment, use 1 for positive and 0 for negative: ‘What a great cast for this movie. The timing was excellent and there were so many clever lines, several times I was still laughing minutes after they were delivered. I found Manna from Heaven to have some surprising moments and while there were things I was thinking would happen, the way they came together was anything but predictable. This movie is about hope and righting wrongs. I left the theater feeling inspired to do the right thing. Bravo to the Five Sisters’. Please only return the label.”}
```

GPT Output:

1

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Disclosure

Preprint notice: A preliminary version of this manuscript was made publicly available on arXiv during the drafting phase [36].

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

No funding was received for this research.

References

- [1] A. Gasparetto, M. Marcuzzo, A. Zangari, and A. Albarelli, “A Survey on Text Classification Algorithms: From Text to Predictions,” *Information* 13, no. 2 (2022): 83, <https://doi.org/10.3390/info13020083>.
- [2] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, “Deep Learning Based Text Classification: A Comprehensive Review,” (2020), <https://arxiv.org/abs/2004.03705>.
- [3] B. Settles, *Active Learning Literature Survey*, Technical Report 1648 (University of Wisconsin–Madison, Computer Sciences, 2009).
- [4] A. Ratner, S. H. Bach, H. Ehrenberg, J. Fries, S. Wu, and C. Ré, “Snorkel: Rapid Training Data Creation With Weak Supervision,” *Proc. VLDB Endow.* 11, no. 3 (2017): 269–282, <https://doi.org/10.14778/3157794.3157797>.
- [5] K. Tomanek and U. Hahn, “Semi-Supervised Active Learning for Sequence Labeling,” in *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Ser. ACL ’09* (Stroudsburg: Association for Computational Linguistics, 2009), 1039–1047, <https://aclanthology.org/P09-1117>.
- [6] E. Strubell, A. Ganesh, and A. McCallum, “Energy and Policy Considerations for Deep Learning in Nlp,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (Florence, Italy, 2019).
- [7] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, “On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (New York, 2021).
- [8] T. B. Brown, B. Mann, N. Ryder, et al., “Language Models Are Few-Shot Learners,” *Advances in Neural Information Processing Systems* 33 (2020): 1877–1901.
- [9] D. Narayanan, A. Phanishayee, K. Shi, X. Chen, and M. Zaharia, “Memory-Efficient Pipeline-Parallel Dnn Training,” *Proceedings of Machine Learning Research* 139 (July 2021): 7937–7947, <https://proceedings.mlr.press/v139/narayanan21a.html>.
- [10] J. W. Rae, S. Borgeaud, T. Cai, et al., “Scaling Language Models: Methods, Analysis & Insights From Training Gopher,” (2021), <https://arxiv.org/abs/2112.11446>.
- [11] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, et al., “A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges,” *IEEE Access* 12 (2024): 26 839–26 874, <https://doi.org/10.1109/access.2024.3365742>.
- [12] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, “Learning Word Vectors for Sentiment Analysis,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (Portland: Association for Computational Linguistics, June 2011), 142–150, <http://www.aclweb.org/anthology/P11-1015>.
- [13] X. Zhang, J. Zhao, and Y. LeCun, “Character-Level Convolutional Networks for Text Classification,” *Advances in Neural Information Processing Systems* 28 (2015): 649–657.
- [14] J. A. Team, “Jigsaw Toxic Comment Classification Challenge Dataset,” (2018), <https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data>.
- [15] F. Sebastiani, “Machine Learning in Automated Text Categorization,” *ACM Computing Surveys* 34, no. 1 (2002): 1–47, <https://doi.org/10.1145/505282.505283>.
- [16] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: Sentiment Classification Using Machine Learning Techniques,” in *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10* (Stroudsburg: Association for Computational Linguistics, 2002), 79–86.
- [17] I. Androutsopoulos, J. Koutsias, V. Chandrinou, and C. D. Spyropoulos, “An Experimental Comparison of Naive Bayesian and Keyword-Based Anti-Spam Filtering With Personal E-Mail Messages,” in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Athens, Greece: ACM, 2000), 160–167.
- [18] D. D. Lewis and W. A. Gale, “A Sequential Algorithm for Training Text Classifiers,” in *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and*

- Development in Information Retrieval* (Dublin, Ireland: Springer-Verlag New York, Inc, 1994), 3–12.
- [19] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. Brown, “Text Classification Algorithms: A Survey,” *Information* 10, no. 4 (2019): 150, <https://doi.org/10.3390/info10040150>.
 - [20] Y. Zhang and B. C. Wallace, “A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification,” (2015), <https://arxiv.org/abs/1510.03820>.
 - [21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding,” (2018), <https://arxiv.org/abs/1810.04805>.
 - [22] Y. Liu, M. Ott, N. Goyal, et al., “Roberta: A Robustly Optimized Bert Pretraining Approach,” (2019), <https://arxiv.org/abs/1907.11692>.
 - [23] OpenAI, “Gpt-4 Technical Report,” (2023), <https://arxiv.org/abs/2303.08774>.
 - [24] OpenAI, “Gpt-4o System Card,” (2024), <https://arxiv.org/abs/2410.21276>.
 - [25] G. Izacard and E. Grave, “Leveraging Passage Retrieval With Generative Models for Open Domain Question Answering,” in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, ed. P. Merlo, J. Tiedemann, and R. Tsarfaty (Association for Computational Linguistics, April 2021), 874–880, <https://aclanthology.org/2021.eacl-main.74/>.
 - [26] T. Goyal, J. J. Li, and G. Durrett, “News Summarization and Evaluation in the Era of Gpt-3,” (2022), <https://arxiv.org/abs/2209.12356>.
 - [27] M. Chen, J. Tworek, H. Jun, et al., “Evaluating Large Language Models Trained on Code,” (2021), <https://arxiv.org/abs/2107.03374>.
 - [28] D. Angluin, “Queries and Concept Learning,” *Machine Learning* 2, no. 4 (1988): 319–342, <https://doi.org/10.1007/bf00116828>.
 - [29] H. S. Seung, M. Oppen, and H. Sompolinsky, “Query by Committee,” in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, Ser. COLT ’92* (New York: Association for Computing Machinery, 1992), 287–294, <https://doi.org/10.1145/130385.130417>.
 - [30] D. Cohn, L. Atlas, and R. Ladner, “Improving Generalization With Active Learning,” *Machine Learning* 15, no. 2 (1994): 201–221, <https://doi.org/10.1007/bf00993277>.
 - [31] K. Brinker, “Incorporating Diversity in Active Learning With Support Vector Machines,” in *Proceedings of the Twentieth International Conference on International Conference on Machine Learning, Ser. ICML’03* (Washington: AAAI Press, 2003), 59–66.
 - [32] O. Sener and S. Savarese, “Active Learning for Convolutional Neural Networks: A Core-Set Approach,” in *International Conference on Learning Representations* (Vancouver, Canada, 2018), <https://openreview.net/forum?id=H1aIuk-RW>.
 - [33] B. Settles and M. Craven, “An Analysis of Active Learning Strategies for Sequence Labeling Tasks,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, ed. M. Lapata and H. T. Ng (Honolulu: Association for Computational Linguistics, October 2008), 1070–1079, <https://aclanthology.org/D08-1112/>.
 - [34] S. Tong and D. Koller, “Support Vector Machine Active Learning With Applications to Text Classification,” *Journal of Machine Learning Research* 2001 (2002): 45–66.
 - [35] A. J. Joshi, F. Porikli, and N. Papanikolopoulos, “Multi-Class Active Learning for Image Classification,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition* (Miami, 2009), 2372–2379.
 - [36] Y. Zhang and S. Takada, “Applying Llms to Active Learning: Towards Cost-Efficient Cross-Task Text Classification Without Manually Labeled Data,” (2025), <https://arxiv.org/abs/2502.16892>.

Copyright © 2025 Yejian Zhang and Shingo Takada. International Journal of Intelligent Systems published by John Wiley & Sons Ltd. This is an open access article under the terms of the Creative Commons Attribution License (the “License”), which permits use, distribution and reproduction in any medium, provided the original work is properly cited. Notwithstanding the ProQuest Terms and Conditions, you may use this content in accordance with the terms of the License. <https://creativecommons.org/licenses/by/4.0/>